Database Cheat Sheet

Suresh Podeti

3 min read · Nov 9, 2023

Photo by Markus Winkler on Unsplash

Structured

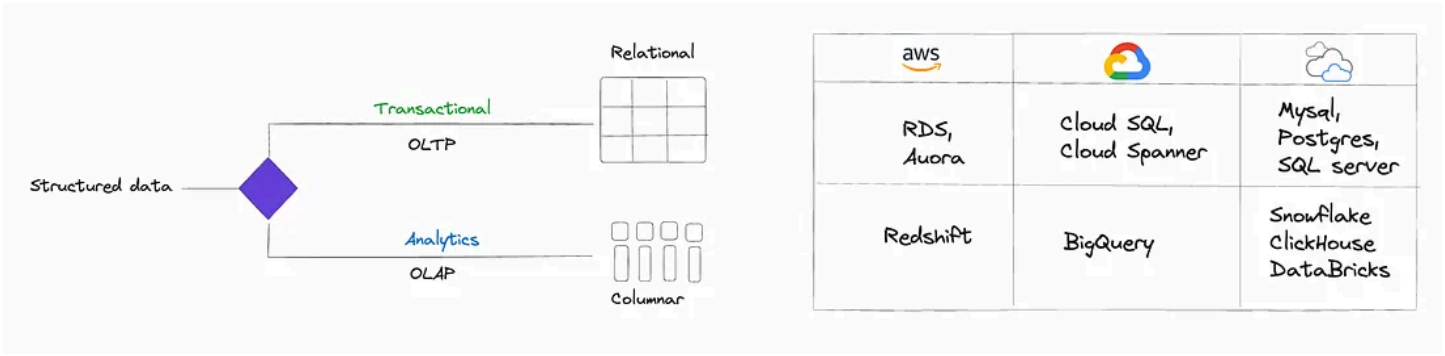


Fig 1.0: Structured databases

Aurora is tailored for **high performance for Mysql, and PostgresDB**, while RDS is a more versatile service supporting various database engines.

Semi-Structured

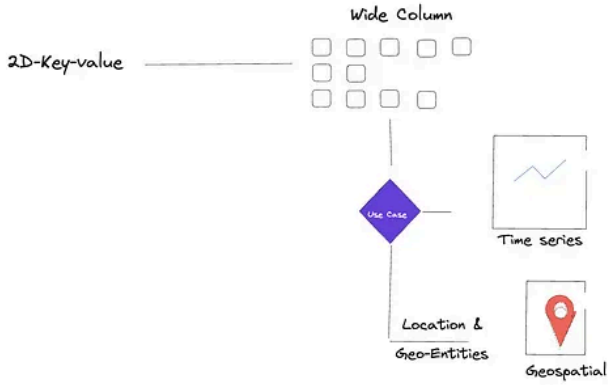
Dictionary

Fig 2.0: Dictionary based databases

While **ScyllaDB's** data model is wide-column, you can think of it as a superset that includes key-value access. Each row in a ScyllaDB table has a primary key, and you can perform efficient lookups based on this key, similar to how key-value stores operate. This makes ScyllaDB suitable for use cases where key-value access patterns are essential.

If you need **advanced data structures, persistence, and more features**, Redis might be the better choice. For simple, high-performance caching, Memcached could be sufficient.

2-D-Key-value



aws		
Keyspaces	BigTable	Cassandra HBase ScyllaDB
TimeStream	BigTable BigQuery	ScyllaDB InfluxDB
Keyspaces	BigTable BigQuery	PostGIS MongoDB (GeoJSON)

Fig 3.0: 2D Key-value databases

PostGIS is an extension for PostgreSQL, a relational database system. The PostGIS introduces a new data type called “geometry” to handle spatial data, which can be stored in a column within a PostgreSQL table.

While PostGIS extends PostgreSQL to handle spatial data effectively, it doesn’t change the underlying structure of PostgreSQL, which remains a relational database.

Similar explanation goes for mongoDB.

Geospatial data often comes in varied and complex formats. **Wide-column stores allow for flexible schema design, where each row in the database can have a different set of columns**. This flexibility is advantageous when dealing with diverse geospatial datasets.

Entity-Relationships

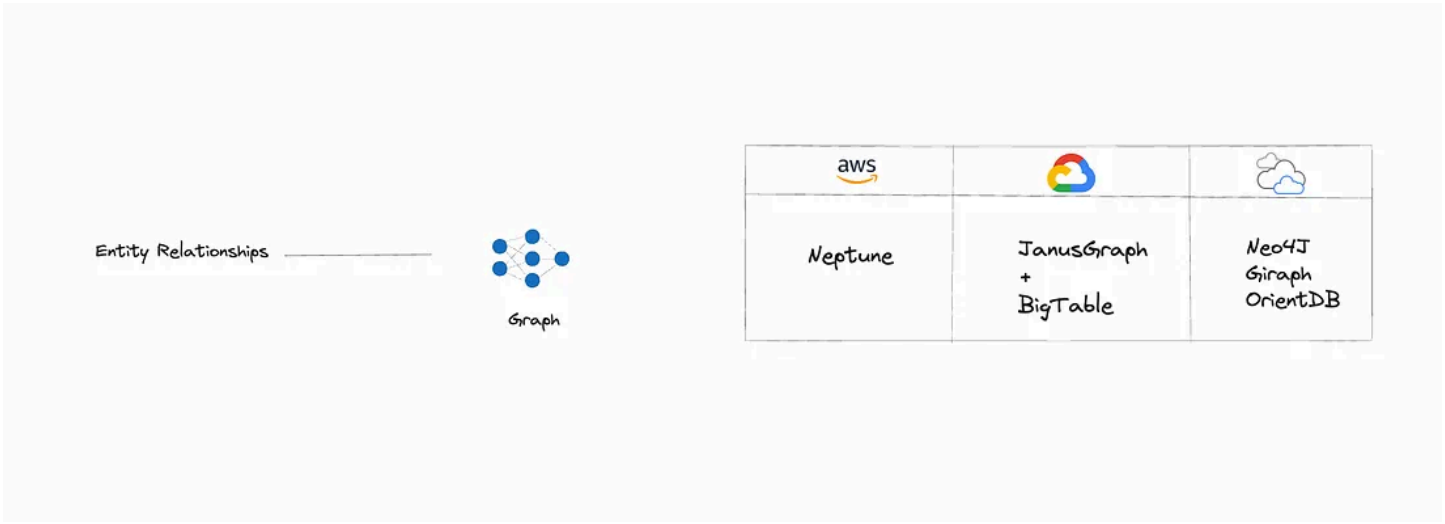


Fig 4.0: Entity-Relationships databases

Nested Objects (XML, JSON)

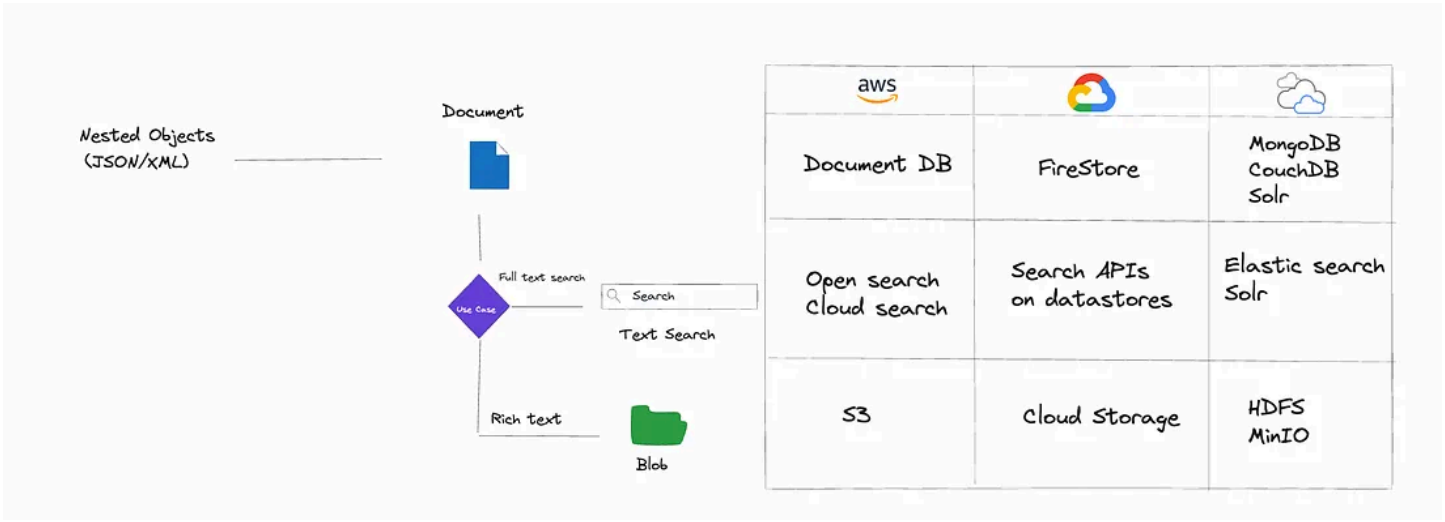


Fig 5.0: Nested objects databases

Choose MongoDB for **versatile, scalable database with powerful querying capabilities, especially when dealing with complex data structures.** Opt for CouchDB if you prioritise a centralised, fault tolerant system that excels in simplicity and ease of replication, making it suitable for distributed scenarios.

Unstructured

Unstructured data, like text messages, pictures, and random notes, cannot be neatly fitted into database tables like structured data can.

Fig x.0: Unstructured data formats

Fig x.0: Unstructured databases

Hadoop is a **distributed storage and processing framework** for big data, while MinIO is an object storage server designed for cloud-native environments. Choose Haddop for large-scale distributed processing and storage. Opt for MinIO if you need scalable object storage with a focus on simplicity and compatibility with cloud native applications.

Database | Choose Database | Distributed Systems | Software Architect | Software Architecture

Edit profile